Abschlussprüfung Sommer 2024

Belegsatz

Fachinformatiker/Fachinformatikerin
Anwendungsentwicklung
1201

Teil 2 der Abschlussprüfung

Der Belegsatz ist Grundlage für beide Prüfungsbereiche.

1. Planen eines Softwareproduktes
2. Entwicklung und Umsetzung von Algorithmen

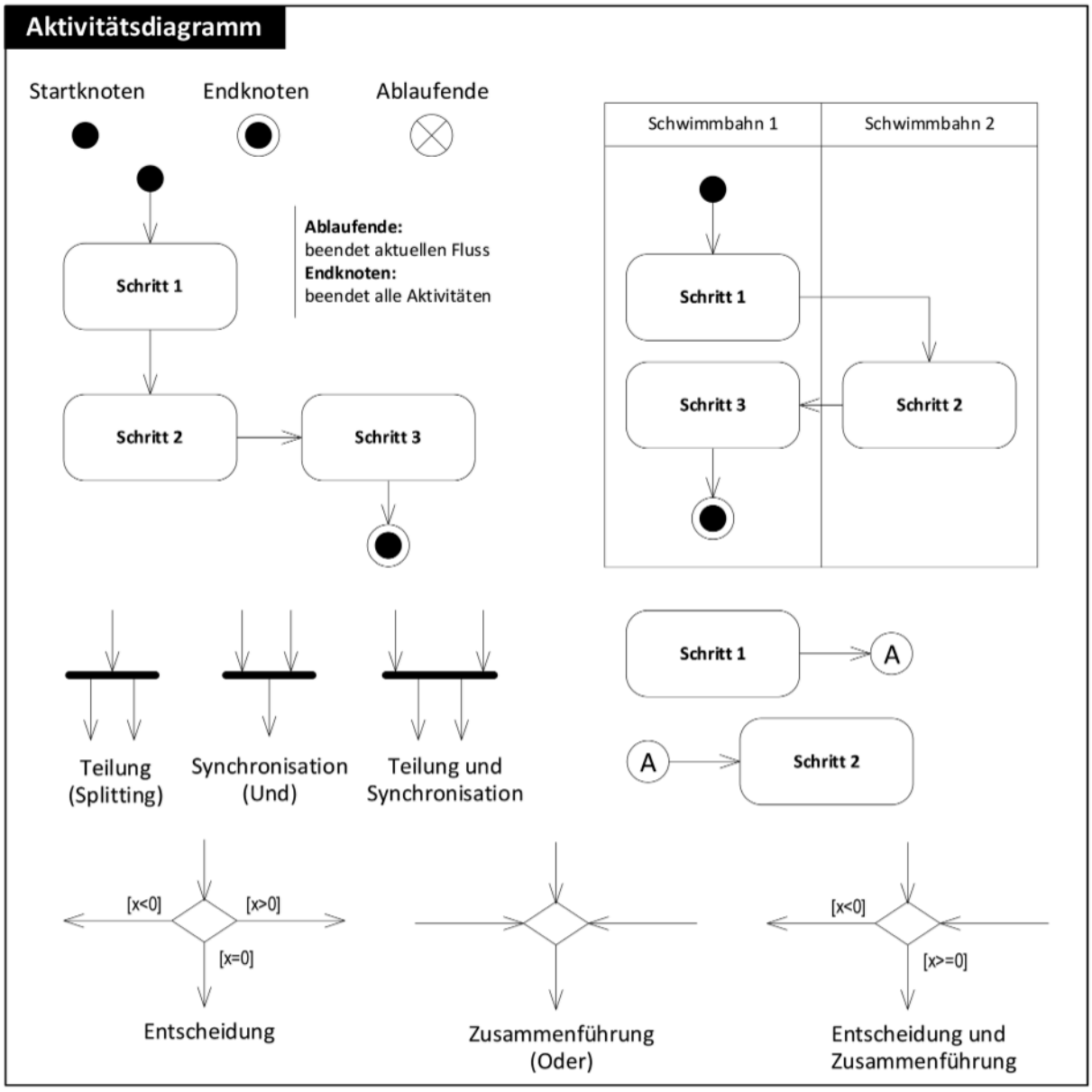
	Seite
Entity Relationship Modell (ER-Modell)	2
UML-Aktivitätsdiagramm	2
UML-Sequenzdiagramm	3
UML-Klassendiagramm	4
UML-Anwendungsfalldiagramm	4
Pseudocode, prozedurale Programmierung	5
SQL-Syntax (Auszug)	6/7

Gemeinsame Prüfungsaufgaben der Industrie- und Handelskammern. Dieser Aufgabensatz wurde von einem überregionalen Ausschuss, der entsprechend § 40 Berufsbildungsgesetz zusammengesetzt ist, beschlossen. Hinweis: Im Interesse einer besseren Lesbarkeit wird in der Aufgabenstellung und in den Angaben zur Aufgabenstellung nur die männliche Form (generisches Maskulinum) verwendet. Die verkürzte Sprachform beinhaltet keine Wertung und die gewählten männlichen Formulierungen gelten uneingeschränkt auch für die weiteren Geschlechter.
Die Vervielfältigung, Verbreitung und öffentliche Wiedergabe der Prüfungsaufgaben und Lösungen ist nicht gestattet. Zuwiderhandlungen werden zivil- und strafrechtlich (§§ 97 ff., 106 ff. UrhG) verfolgt. – © ZPA Nord-West 2024 – Alle Rechte vorbehalten!

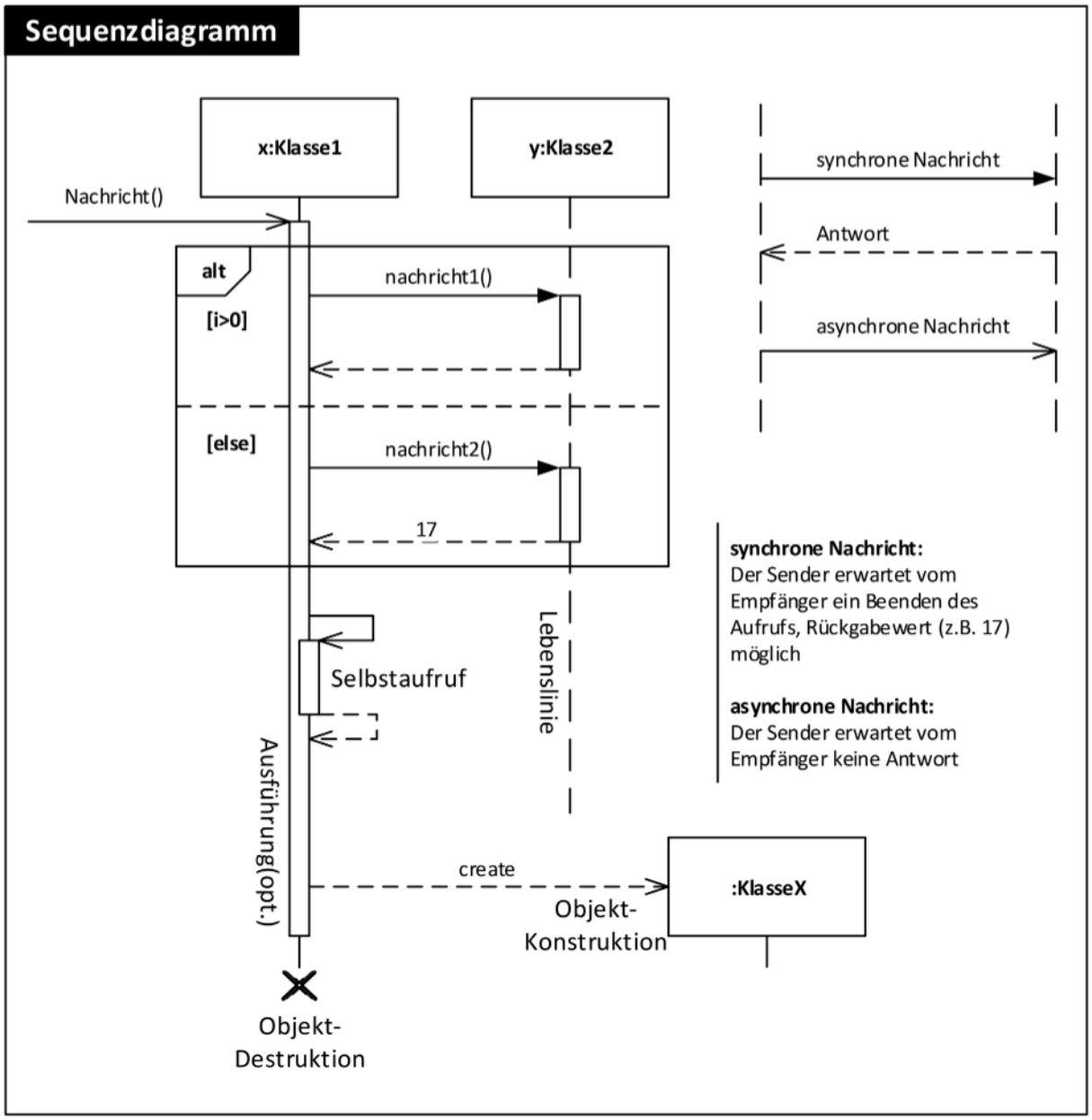
Entity Relationship Modell (ER-Modell)

	Entität
	Beziehung
	Attribut

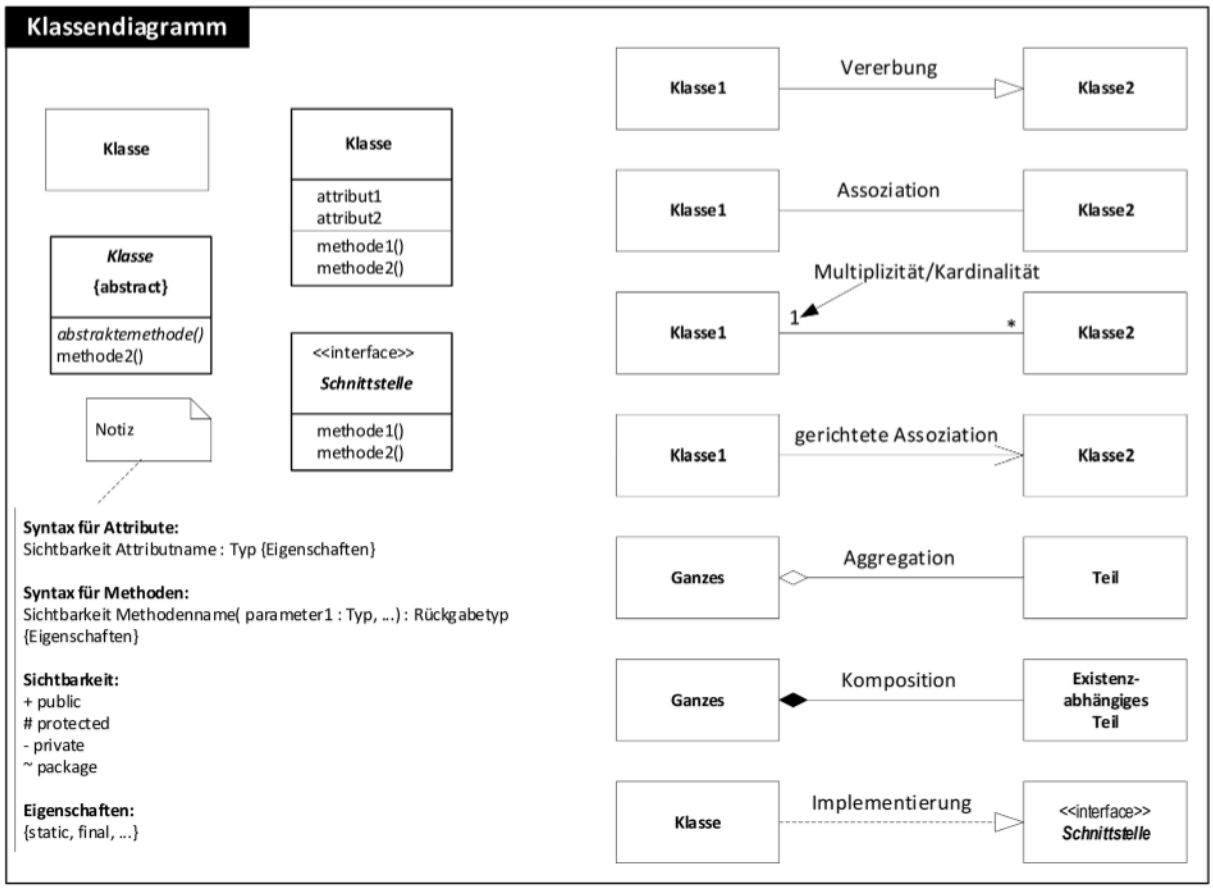
UML-Aktivitätsdiagramm



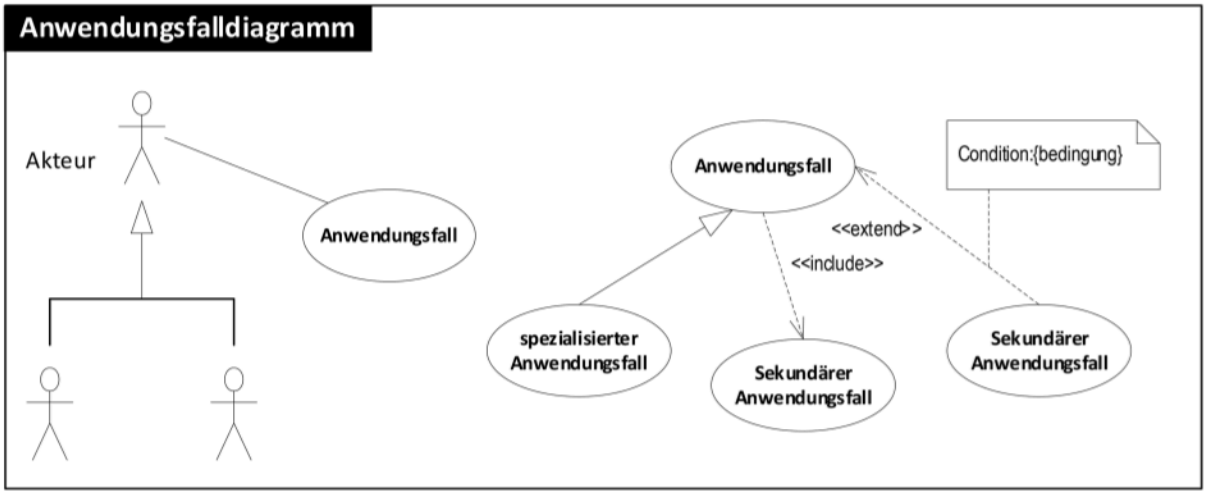
UML-Sequenzdiagramm



UML-Klassendiagramm



UML-Anwendungsfalldiagramm



Pseudocode, prozedurale Programmierung

Schriftart: Courier New (12)

Pseudocode	Beschreibung
wenn [Bedingung] **dann** [Anweisung] **sonst** [Anweisung] **Ende wenn**	Verzweigung
zähle [Variable] **von** [Startwert] **bis** [Endwert] [Schleifeninhalt] **Ende zähle**	Zählschleife (Iteration)
solange [Bedingung] [Schleifeninhalt] **Ende solange**	Kopfgesteuerte Schleife
wiederhole [Schleifeninhalt] **solange** < Bedingung >	Fußgesteuerte Schleife
:=	Zuweisung
=; <; >; <=; >=; < >	Vergleiche
integer	
double	
string	
array	

SQL-Syntax (Auszug)

Syntax	Beschreibung
Tabelle	
CREATE TABLE Tabellenname(Spaltenname < DATENTYP >, Primärschlüssel, Fremdschlüssel)	Erzeugt eine neue leere Tabelle mit der beschriebenen Struktur
ALTER TABLE Tabellenname **ADD COLUMN** Spaltenname Datentyp **DROP COLUMN** Spaltenname Datentyp **ADD FOREIGN KEY**(Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Änderungen an einer Tabelle: Hinzufügen einer Spalte Entfernen einer Spalte Definiert eine Spalte als Fremdschlüssel
CHARACTER	Textdatentyp
DECIMAL	Numerischer Datentyp (Festkommazahl)
DOUBLE	Numerischer Datentyp (Doppelte Präzision)
INTEGER	Numerischer Datentyp (Ganzzahl)
DATE	Datum (Format DD.MM.YYYY)
PRIMARY KEY (Spaltenname)	Erstellung eines Primärschlüssels
FOREIGN KEY (Spaltenname) **REFERENCES** Tabellenname(Primärschlüsselspaltenname)	Erstellung einer Fremdschlüssel-Beziehung
DROP TABLE Tabellenname	Löscht eine Tabelle
Befehle, Klauseln, Attribute	
SELECT * | Spaltenname1 [, Spaltenname2, …]	Wählt die Spalten einer oder mehrerer Tabellen, deren Inhalte in die Liste aufgenommen werden sollen; alle Spalten (*) oder die namentlich aufgeführten
FROM	Name der Tabelle oder Namen der Tabellen, aus denen die Daten der Ausgabe stammen sollen
SELECT … **FROM** … **(SELECT …** **FROM …** **WHERE …) AS** tbl **WHERE** …	Unterabfrage (subquery), die in eine äußere Abfrage eingebettet ist. Das Ergebnis der Unterabfrage wird wie eine Tabelle – hier mit Namen "tbl" – behandelt.
SELECT **DISTINCT**	Eliminiert Redundanzen, die in einer Tabellen auftreten können, Werte werden jeweils nur einmal angezeigt.
JOIN / INNER JOIN	Liefert nur die Datensätze zweier Tabellen, die gleiche Datenwerte enthalten
LEFT JOIN / LEFT OUTER JOIN	Liefert von der erstgenannten (linken) Tabelle alle Datensätze und von der zweiten Tabelle jene, deren Datenwerte mit denen der ersten Tabelle übereinstimmen
RIGHT JOIN / RIGHT OUTER JOIN	Liefert von der zweiten (rechten) Tabelle alle Datensätze und von der ersten Tabelle jene, deren Datenwerte mit denen der zweiten Tabelle übereinstimmen
WHERE	Bedingung, nach der Datensätze ausgewählt werden sollen
WHERE EXISTS (subquery **)** **WHERE NOT EXISTS (** subquery **)**	Die Bedingungen EXISTS prüft, ob die Suchbedingung einer Unterabfrage mindestens eine Zeile zurückliefert. NOT EXIST negiert die Bedingung.
WHERE ... IN (subquery **)** **WHERE NOT... IN (** subquery **)**	Der Wert des Datenfelds ist in der auswählten Menge vorhanden. Der Wert des Datenfelds ist in der auswählten Menge nicht vorhanden.
GROUP BY Spaltenname1 [,Spaltenname2, …]	Gruppierung (Aggregation) nach Inhalt des genannten Feldes
ORDER BY Spaltenname1 [,Spaltenname2, …] **ASC | DESC**	Sortierung nach Inhalt des genannten Feldes oder der genannten Felder ASC: aufsteigend; DESC: absteigend

Syntax	Beschreibung
Datenmanipulation	
DELETE FROM Tabellenname	Löschen von Datensätzen in der genannten Tabelle
UPDATE Tabellenname **SET**	Aktualisiert Daten in Feldern einer Tabelle
INSERT INTO Tabellenname[(spalte1, spalte2, ...)] **VALUES (**Wert für Spalte 1 [, Wert für Spalte 2, …]**)** oder **SELECT** … **FROM** … **WHERE**	Fügt Datensätze in die genannte Tabelle, die entweder mit festen Werten belegt oder Ergebnis eines SELECT-Befehls sind
Berechtigungen kontrollieren	
CREATE Benutzer | Rolle **IDENTIFIED BY** 'Passwort'	Erzeugt einen neuen Benutzer oder eine neue Rolle mit einem Passwort
GRANT Recht | Rolle **ON** *.* | Datenbank.* | Datenbank.Objekt **TO** Benutzer | Rolle [WITH GRANT OPTION]	Weist einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt zu Weist einem Benutzer eine Rolle zu
REVOKE Rechte | Rollen **ON** *.* | Datenbank.* | Datenbank.Objekt **FROM** Benutzer | Rolle	Entzieht einem Benutzer oder einer Rolle ein Recht auf ein bestimmtes Datenbank-Objekt Entzieht einem Benutzer eine Rolle
Aggregatfunktionen	
AVG(Spaltenname**)**	Ermittelt das arithmetische Mittel aller Werte im angegebenen Feld
COUNT(Spaltenname **| *)**	Ermittelt die Anzahl der Datensätze mit Nicht-NULL-Werten im angegebenen Feld oder alle Datensätze der Tabelle (dann mit Operator *)
SUM(Spaltenname **|** Formel**)**	Ermittelt die Summe aller Werte im angegebenen Feld oder der Formelergebnisse
MIN(Spaltenname **|** Formel**)**	Ermittelt den kleinsten aller Werte im angegebenen Feld
MAX (Spaltenname | Formel**)**	Ermittelt den größten aller Werte im angegebenen Feld
Funktionen	
LEFT(Zeichenkette, Anzahlzeichen**)**	Liefert *Anzahlzeichen* der Zeichenkette von links.
RIGHT(Zeichenkette, Anzahlzeichen**)**	Liefert *Anzahlzeichen* der Zeichenkette von rechts.
CURRENT	Liefert das aktuelle Datum mit der aktuellen Uhrzeit
CONVERT(time,[DatumZeit**])**	Liefert die Uhrzeit aus einer DatumZeit-Angabe
DATE(Wert**)**	Wandelt einen Wert in ein Datum um
DAY(Datum**)**	Liefert den Tag des Monats aus dem angegebenen Datum
MONTH(Datum**)**	Liefert den Monat aus dem angegebenen Datum
TODAY	Liefert das aktuelle Datum
WEEKDAY(Datum**)**	Liefert den Tag der Woche aus dem angegebenen Datum als Text
YEAR(Datum**)**	Liefert das Jahr aus dem angegebenen Datum
DATEADD(Datumsteil, Intervall, Datum)	Fügt einem Datum ein Intervall (ausgedrückt in den unter Datumsteil angegebenen Einheiten) hinzu
DATEDIFF(Datumsteil, Anfangsdatum, Enddatum) Datumsteile: **DAY, MONTH, YEAR**	Liefert Enddatum-Startdatum (ausgedrückt in den unter Datumsteil angegebenen Einheiten)
Operatoren	
UNION ALL	Verkettung die Ergebnismengen von zwei Abfragen
UNION	Wie UNION, eliminiert doppelte Zeilen
AND	Logisches UND
LIKE	Überprüfung von Text auf Gleichheit, wenn Platzhalter ("regular expressions") eingesetzt werden.
NOT	Logische Negation
OR	Logisches ODER
IS NULL	Überprüfung auf NULL
=	Test auf Gleichheit
>, >=, <, <=, < >	Test auf Ungleichheit
*	Multiplikation
/	Division
+	Addition, positives Vorzeichen
-	Subtraktion, negatives Vorzeichen

Stand 2021-09-30